HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

Divyanshu Rawat 36141

Task 4

ATM system using Tag

NEAR FIELD COMMUNICATION(NFC)

Prof. Dr. Markus Pfeil

Content List

Picture List

1. Introduction

This report explores the development of an Arduino-based ATM Banking System, featuring NFC card authentication, PIN verification, account balance inquiry, and money withdrawal functionalities. Utilizing an Arduino Nano, PN532 NFC module, and a 4x4 membrane keypad, the system offers a cost-effective and replicable solution for basic banking operations. Through this project, we demonstrate how Arduino technology can create a secure and convenient banking experience, simulating the core features of a real-world ATM. The report covers hardware components, software design, programming logic, and user experience, providing insights into microcontroller-based banking systems and ATM technology.

2. Basic Theory

2.1 Near Field Communication

Near Field Communication (NFC) is a short-range wireless communication technology that allow the device to exchange data over a short distance. It operates on the principles of electromagnetic induction and radio frequency identification (RFID).

The basic theory Of NFC involves two components : the NFC reader and NFC tag or device. The NFC reader can be a smartphone, tablet, or any other device equipped with NFC capabilities, while the NFC tag can be passive tag, such as sticker or card, or an active device, such as another smartphone or an NFC-enabled electronic device.

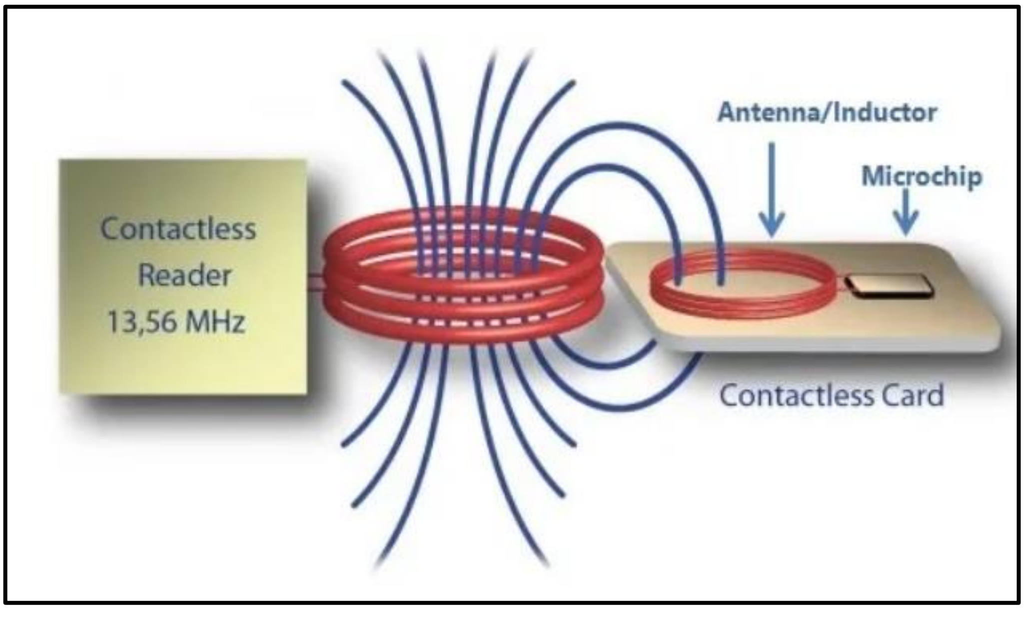


Figure 1

Difference Between RFID & NFC

The key distinction between RFIC and NFC lies in their transmission ranges. The RFID is often over longer distance. For example, some regions automatically collect road tolls through RFID. Communication can take place over even longer distances if the RFID tag is equipped with power source.

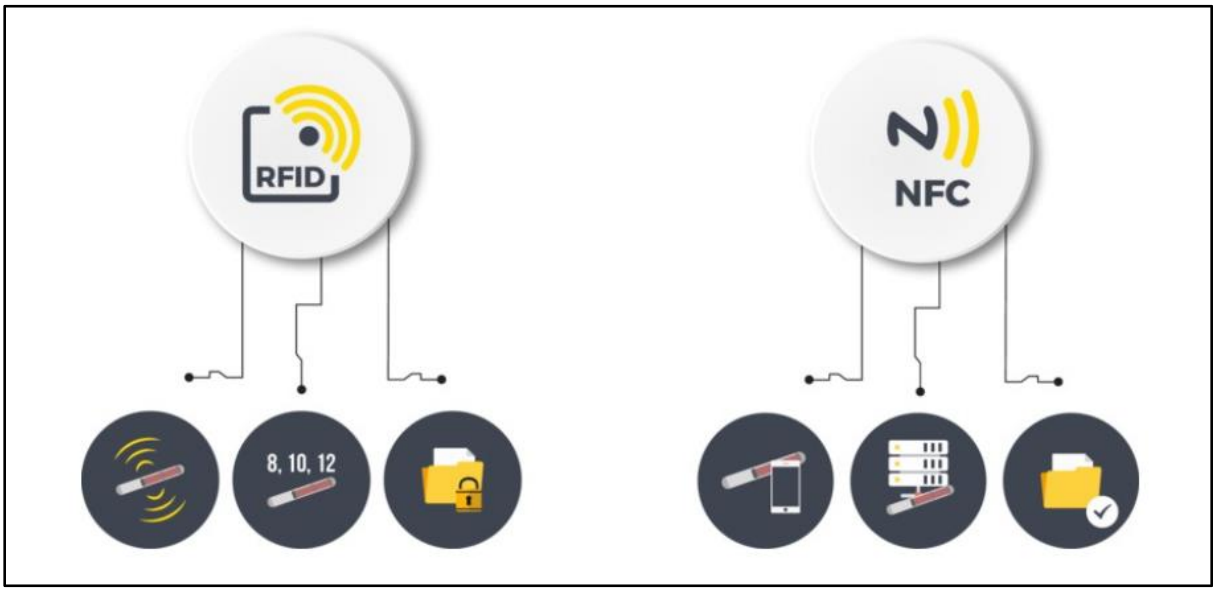


Figure 2

2.2 Arduino Nano

The Arduino Nano is small, compact and versatile microcontroller borad based on the ATmega328P microcontroller. It is prat of the Arduino Family of development boards and is designed for projects that require a small form factor and low power consumption. The Arduino Nano is a popular choice for hobbyists, students and professionals a like due to its simplicity and affordability

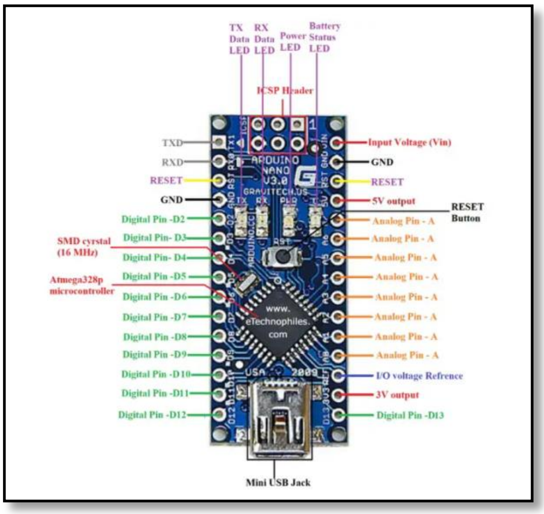


Figure 3

2.3 PN 532 NFC Module

The PN532 module supports both reading and writing NFC data, as well as peer to peer communication between devices. It operates at the 13.56 MHz frequency and is compatible with NFC forum standards.

Communication Interface the module interface with microcontroller or other devices using different communication protocols, such as I2C (inter-integrated Circuits), SPI (Serial Peripheral Interface), and UART

PN532 module typically comes with a built-in antenna, eliminating the need for external antenna connections. This simplifies the integration process and reduces the overall size and complexity of the project.

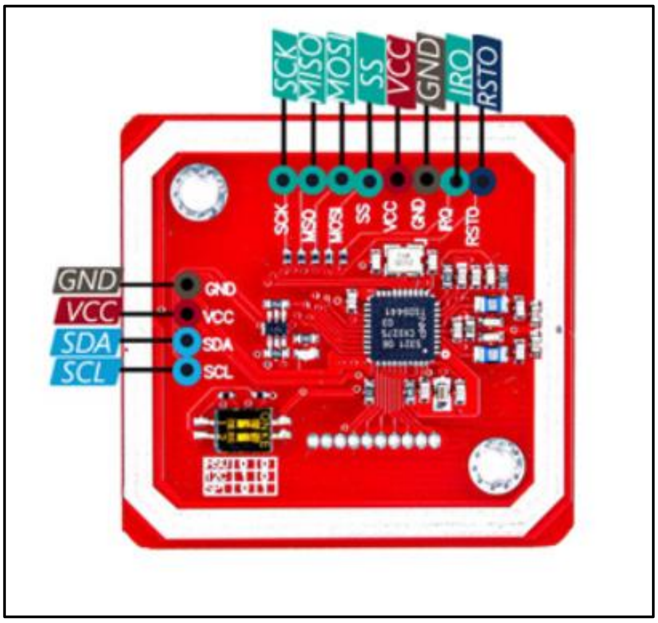


Figure 4

2.4 Mifare Classic NFC Card/Tag

A Mifare Classic NFC card/tag refers to a specific type of NFC tag that utilizes the MiFare Classic technology developed by NXP semiconductors. Mifare Classic is one of the most widely used NFC tag technologies.

Mifare Classic NFC cards/tags operate at the 13,56 MHz frequency, which is the standart frequency for NFC communication. They are compatible with NFC-enabled devices such as smartphones, NFC readers and other NFC-capable devices.

The most common option are 1KB, 4KB, and 7KB or EEPROM storage. The memory is organized into sectors and blocks, allowing for data storage and retrieval.

Mifare Classic NFC cards/tags support both reading and writing pf data. This allows for dynamic interaction with the tag, enabling application such as updating stored values, adding or removing access permissions and personalizing the card/tag content.

2.5 Keypad 20 membrane 4x5 :

The keypad is an input device that allows users to enter numeric and other characters.

It consists of a grid of 20 keys arranged in a 4x5 matrix format (4 row and 5 columns). Each key represent a specific character (number 0-9 and additional character like "*", "#", etc).

When a key is pressed, it connect the corresponding row and column, enabling the Arduino to detect which key was pressed by scanning the rows and columns of the keypad matrix.

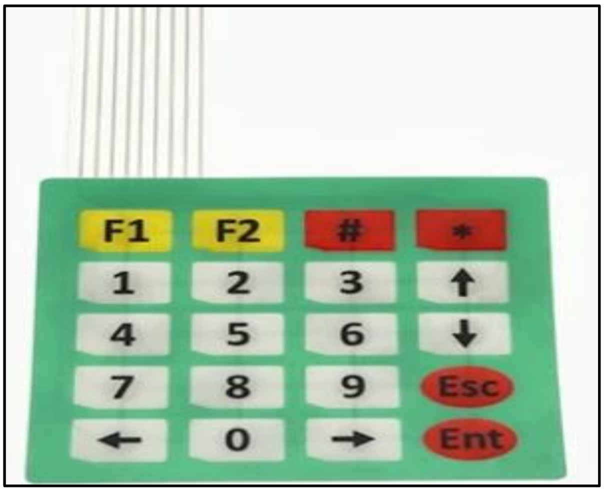


Figure 5

2.6 Relay Tongling 5V DC:

A relay is an electrically operated switch that allows a low-power circuit (like Arduino) to control a high-power circuit (like a motor, lights, or other devices).

The Tongling relay operates with a 5V DC input from the Arduino.

It has two main components: a coil and contacts.

When the coil is energized (by applying 5V), it creates a magnetic field, causing the contacts to close or open, thus turning the connected high-power circuit on or off.

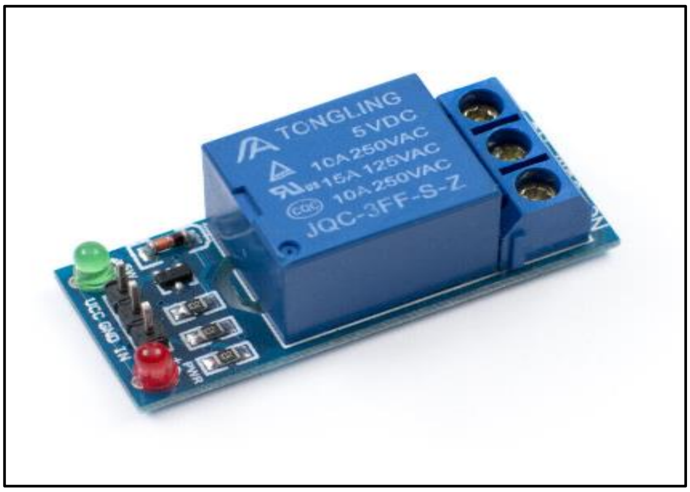


Figure 6

2.7 Buzzer Module (Low Level Trigger):

The buzzer module is an output device that produces an audible sound.

It operates with a low-level trigger, which means it activates when a low voltage or ground is applied to its signal input.

When the Arduino sends a LOW signal to the buzzer module, it generates a sound.

It's commonly used to provide audio feedback or alerts in various projects, including an ATM system to indicate successful authentication or other actions.

In summary, the 20-membrane 4x5 keypad is used to input data (e.g., PIN), the Tongling relay is employed to control high-power devices (like the ATM cash dispenser), and the buzzer module is used for audio feedback in the project. Together, these components enable the creation of a functional and interactive Arduino-based ATM Banking System.

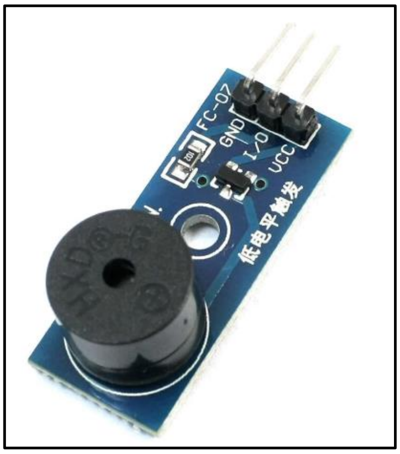
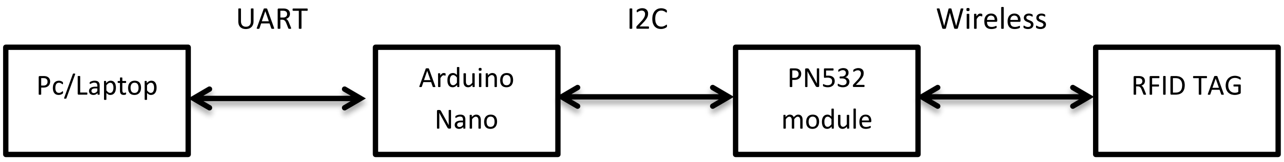


Figure 7

3. Hardware Design and Development

3.1 Block Diagram



Connect the Arduino Nano to laptop using USB cable. This connection allows the Arduino Nano to receive power and establish a serial communication link for programming and data exchange.

Connect the VCC pin of the PN532 module to the 3.3V or 5V pin the Arduino Nano depending on the module voltage requirement. Connect the GND pin of the PN532 module to any GND pin on the Arduino Nano to establish a common ground.

If you prefer to use I2C communication instead of Serial communication, you can connect the SDA(data line) and SCL (clock line) pins of the PN532 module to the corresponding I2c pins on the Arduino Nano (SDA to A4, SCl To A5).

The PN532 module is equipped with an integrated antenna that generates an electromagnetic field. When an RFID tag comes within the range of the PN532 modules electromagnetic field, it is powered by the energy induced in the tag’s antenna.

3.2 Hardware Setup Result

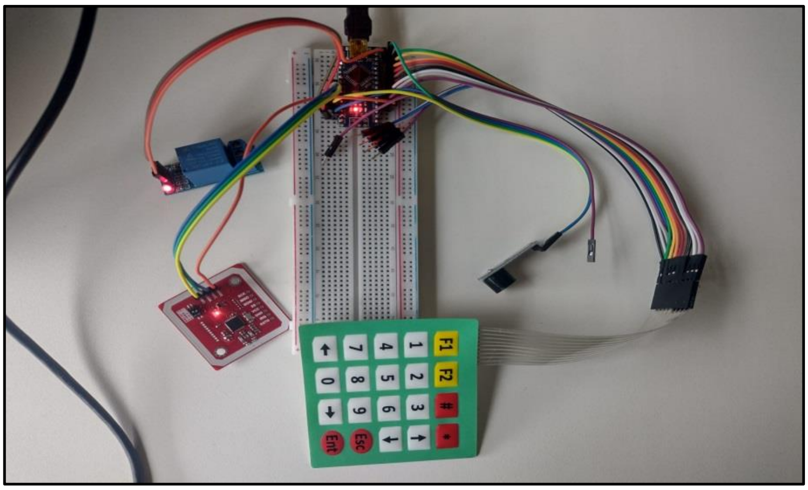


Figure 8

4. Software Design and Development

4.1 Flowchart

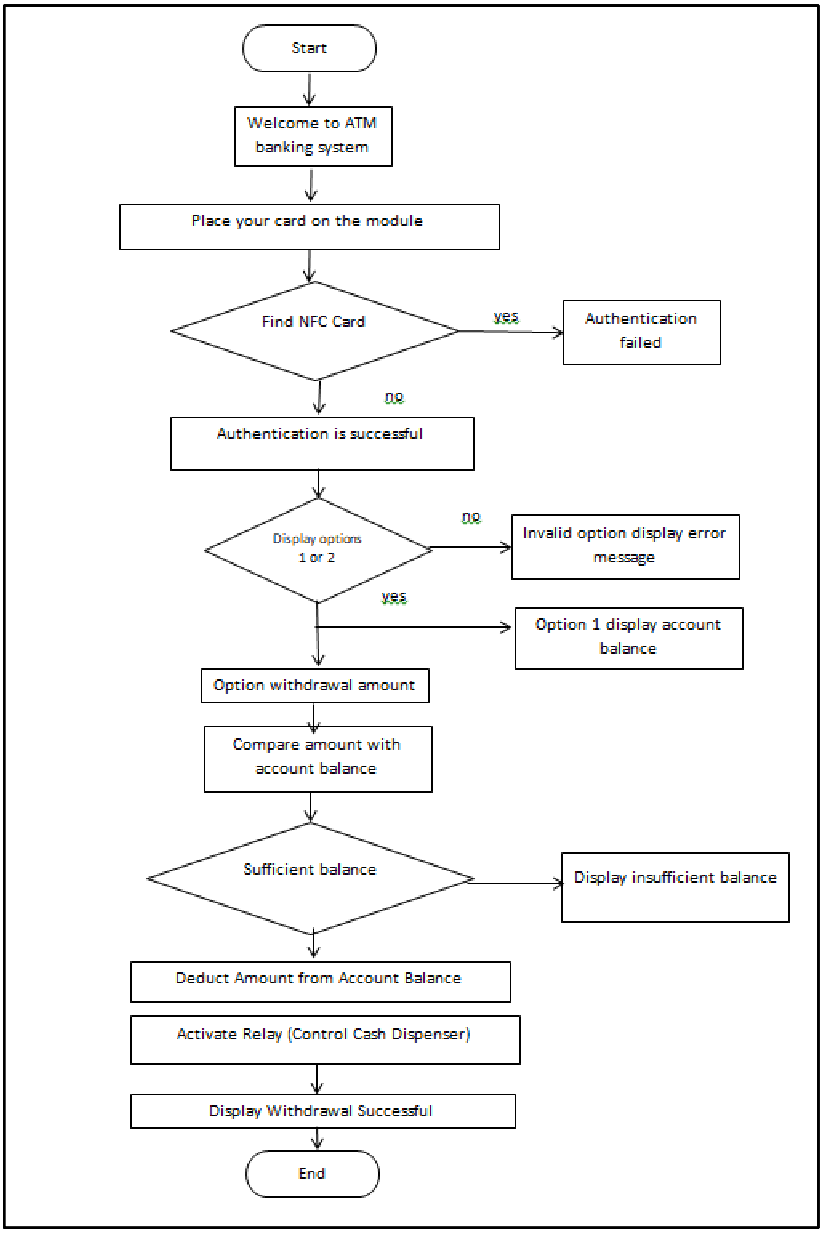


Figure 9

4.2 Software Code

```
#include <Wire.h>
#include <nfc.h>
#include <Keypad.h>

NFC_Module nfc;
```

```
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

int accountBalance = 100;
bool authenticated = false;

const int buzzerPin = 12;
const int relayPin = 11;

void setup(void) {
  Serial.begin(9600);
  nfc.begin();
  Serial.println("MF1S50 Reader Demo From Elechouse!");

  uint32_t versiondata = nfc.get_version();
  if (!versiondata) {
    Serial.print("Didn't find PN53x board");
    while (1);
  }

  Serial.print("Found chip PN5");
  Serial.println((versiondata >> 24) & 0xFF, HEX);
  Serial.print("Firmware ver. ");
  Serial.print((versiondata >> 16) & 0xFF, DEC);
  Serial.print('.');
  Serial.println((versiondata >> 8) & 0xFF, DEC);

  nfc.SAMConfiguration();

  pinMode(buzzerPin, OUTPUT);
  pinMode(relayPin, OUTPUT);

  digitalWrite(buzzerPin, HIGH);
  digitalWrite(relayPin, HIGH);

  Serial.println("Welcome to the ATM Banking System");
  Serial.println("Place your card on the module.");
}
```

```
void loop(void) {
  u8 buf[32], sta;
  unsigned long start_time, end_time;

  if (!authenticated) {
    sta = nfc.InListPassiveTarget(buf);

    if (sta && buf[0] == 4) {
      Serial.print("UUID length: ");
      Serial.println(buf[0], DEC);
      Serial.print("UUID: ");
      nfc.puthex(buf + 1, buf[0]);
      Serial.println();

      u8 key[6] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};

      sta = nfc.MifareAuthentication(0, 5, buf + 1, buf[0], key);
      if (sta) {
        Serial.println("Authentication success!");

        char enteredPIN[5];
        uint8_t index = 0;

        Serial.println("Enter the PIN:");

        while (index < 4) {
          char key = keypad.getKey();
          if (key) {
            enteredPIN[index] = key;
            Serial.print(key);
            index++;
          }
        }
        enteredPIN[4] = '\0';

        u8 pinNumber[16];
        sta = nfc.MifareReadBlock(5, pinNumber);
        if (sta) {
          Serial.println("\nPIN Number read successfully!");

          if (strcmp(enteredPIN, (char*)pinNumber) == 0) {
            Serial.println("Password is correct");

            authenticated = true;

            Serial.println("Authentication successful!");

            // Turn on the buzzer for 3 seconds
```

```
          digitalWrite(buzzerPin, LOW);
          delay(3000);
          digitalWrite(buzzerPin, HIGH);

          Serial.println("Select an option:");
          Serial.println("1. View Account Balance");
          Serial.println("2. Money Withdrawal");
        } else {
          Serial.println("Password is incorrect");
          authenticated = false;
        }
      } else {
        Serial.println("Error reading PIN Number from the block.");
      }
    } else {
      Serial.println("Authentication failed.");
    }
  }
} else {
  char key = keypad.getKey();
  if (key) {
    if (key == '1') {
      Serial.print("Account balance: ");
      Serial.println(accountBalance);
    } else if (key == '2') {
      Serial.println("Enter the withdrawal amount:");
      uint32_t withdrawalAmount = 0;

      while (true) {
        char key = keypad.getKey();
        if (key) {
          if (key >= '0' && key <= '9') {
            withdrawalAmount = withdrawalAmount * 10 + (key - '0');
            Serial.print(key);
          } else if (key == '#') {
            break;
          }
        }
      }

      if (withdrawalAmount <= accountBalance) {
        accountBalance -= withdrawalAmount;
        Serial.print("\nWithdrawn amount: ");
        Serial.println(withdrawalAmount);
        Serial.print("Remaining balance: ");
        Serial.println(accountBalance);

        // Turn on the relay for 1 second
```

```
          digitalWrite(relayPin, LOW);
          delay(1000);
          digitalWrite(relayPin, HIGH);
        } else {
          Serial.println("\nInsufficient account balance!");
        }
      } else {
        Serial.println("Invalid option selected.");
      }
    }
  }

  delay(100);
}
```

5. Result and Analysis

5.1 Serial Monitor Result View

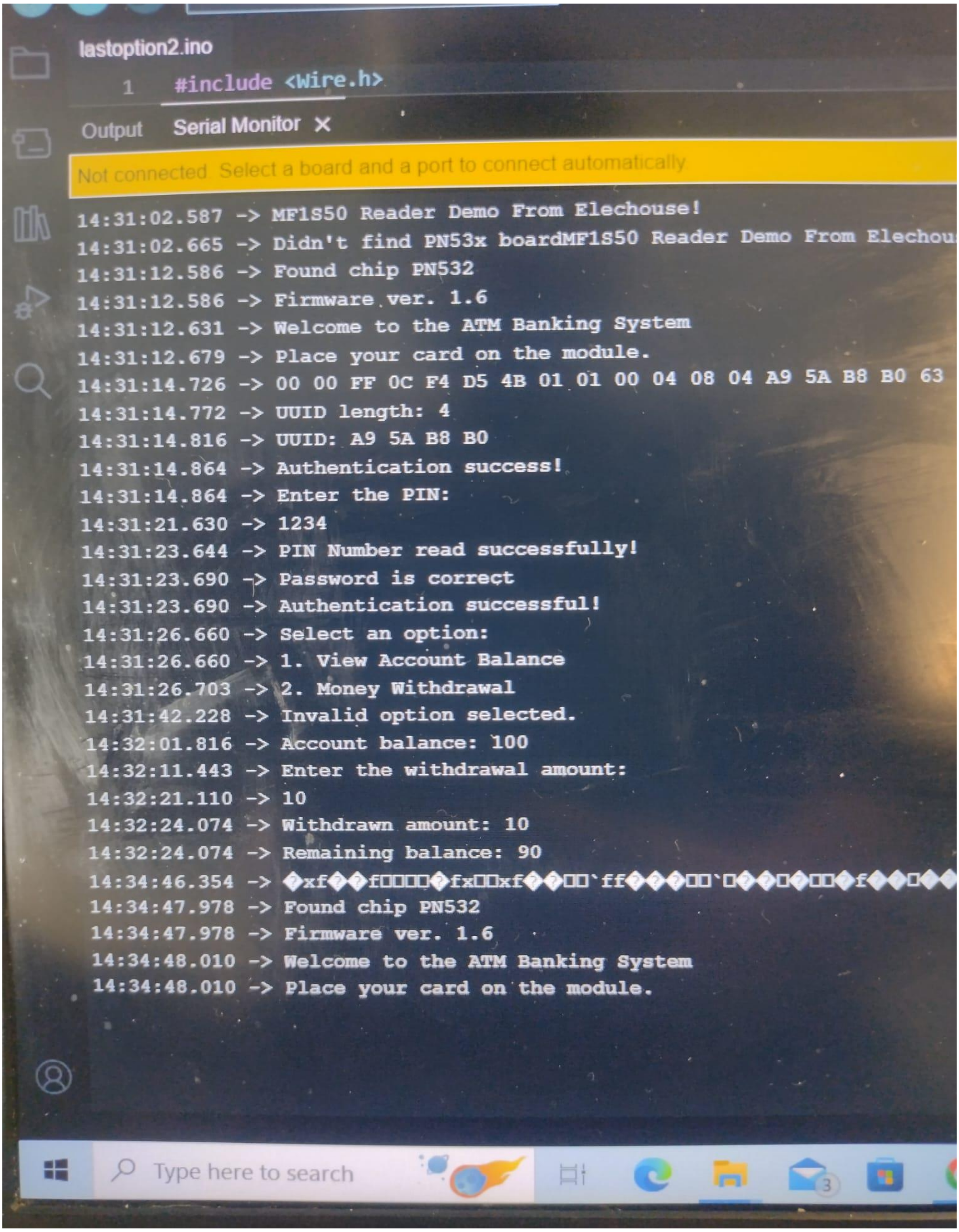


Figure 10

6. Reference

https://learn.adafruit.com/adafruit-pn532-rfidnfc/mifarehttps://learn.adafruit.com/adafruit-pn532-rfid-nfc/mifare

Mifare data shee